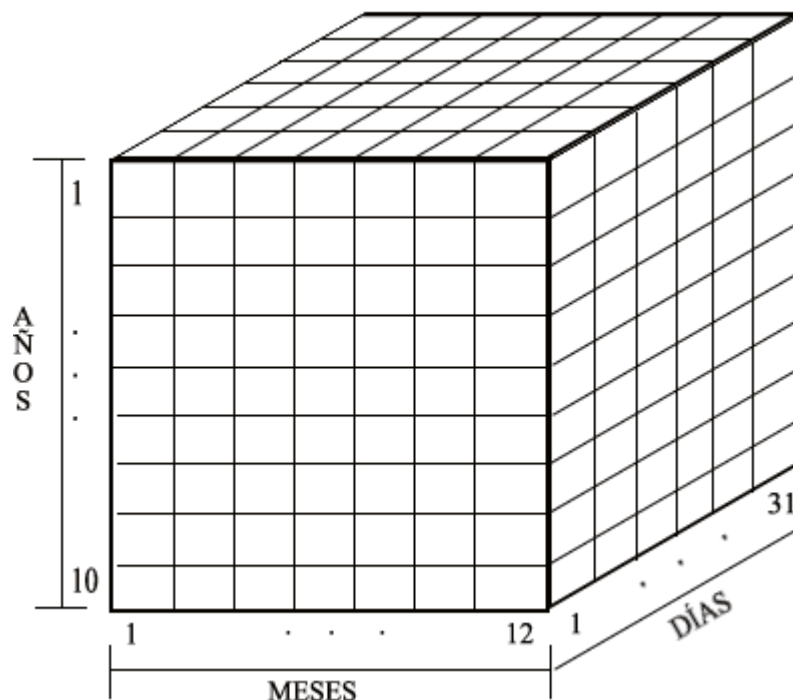


ESTRATEGIAS PARA LA RESOLUCIÓN DE PROBLEMAS CON MATRICES N-DIMENSIONALES

ENUNCIADO

El Instituto del Servicio Meteorológico Nacional le ha proporcionado la siguiente matriz tridimensional que posee registros de las lluvias caídas diariamente durante los últimos 10 años.



Para hacer análisis estadísticos y proyecciones a futuro, se le pide a usted que realice el diagrama de flujo y programa en Pascal para resolver los siguientes requisitos:

- La cantidad de días sin lluvias.
- El año que más llovió.
- El mes que menos llovió y en qué año ocurrió.
- El día que más llovió en cada año y en qué mes ocurrió, almacenando los datos en una matriz de AÑOS x (DÍA; MES; VALOR)

ANÁLISIS

Veamos y analicemos los requerimientos con cuidado para diseñar la estrategia adecuada de resolución. En principio podemos considerar cada ítem como independiente del resto y tratarlo como si fuese un problema individual. Entonces cada ítem a resolver posee una complejidad menor que el problema integral y es más fácil de ser abordado. Los tres primeros ítems son de similar complejidad, pero el cuarto (d) tiene una complejidad mayor porque requiere una búsqueda más exhaustiva de los datos e inclusive también nos obliga a presentar los datos que encontramos en una determinada manera.

Antes de comenzar a resolver es bueno ver cómo recorreremos la matriz de datos. Una matriz N-dimensional es, en definitiva, como cualquier arreglo de datos. Según su cantidad de dimensiones requerirá la misma cantidad de índices para recorrerla completamente. En nuestro caso, la matriz tiene dimensión 3 (años x meses x días) y con tres índices (i, j, k) podremos recorrer todos sus elementos. Llamaremos A a la matriz y $A[i, j, k]$ al elemento en la posición (i) años, (j) meses y (k) días. Así las cosas, nos podremos dedicar a resolver cada ítem individualmente, siempre teniendo en cuenta la necesidad de inicializar variables en el lugar indicado del programa y desarrollando los diagramas de flujo de la manera más óptima.

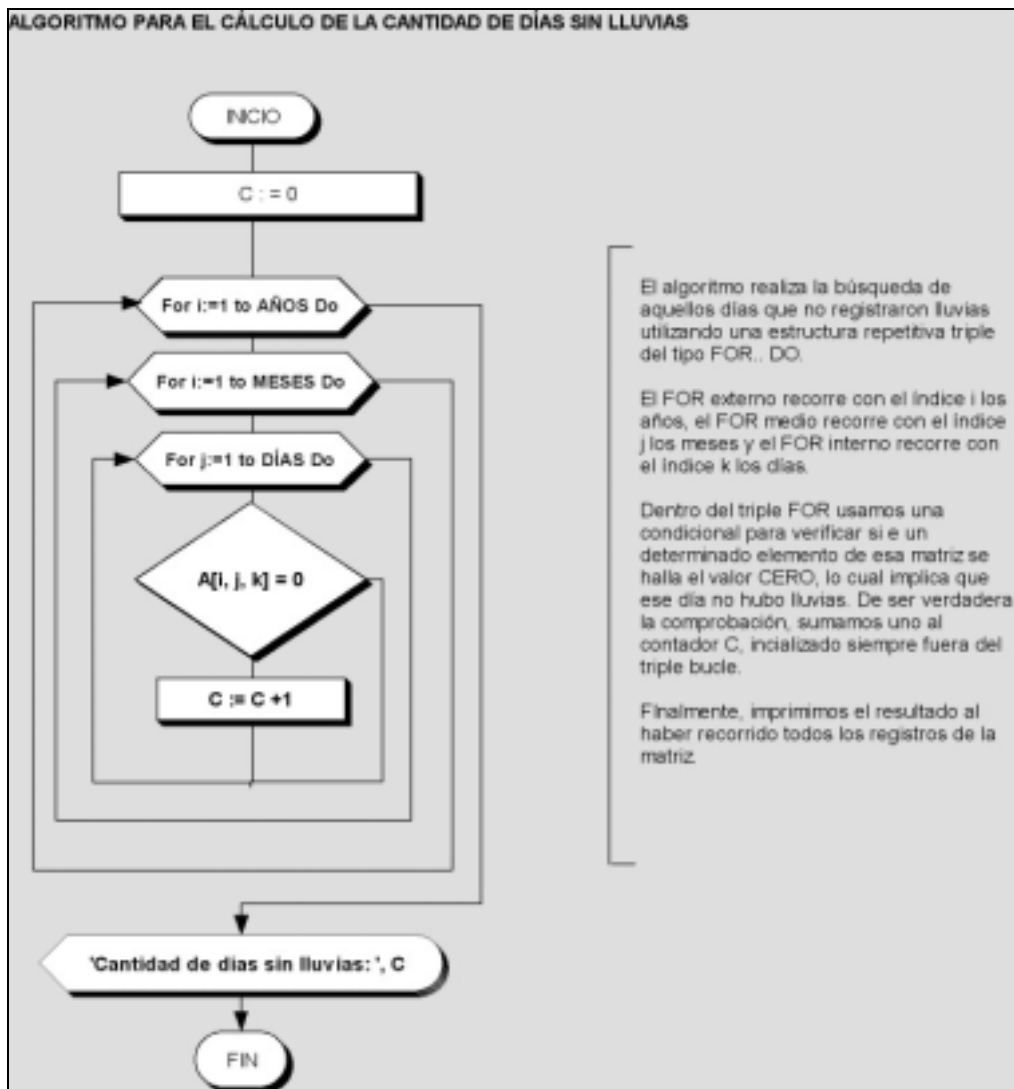
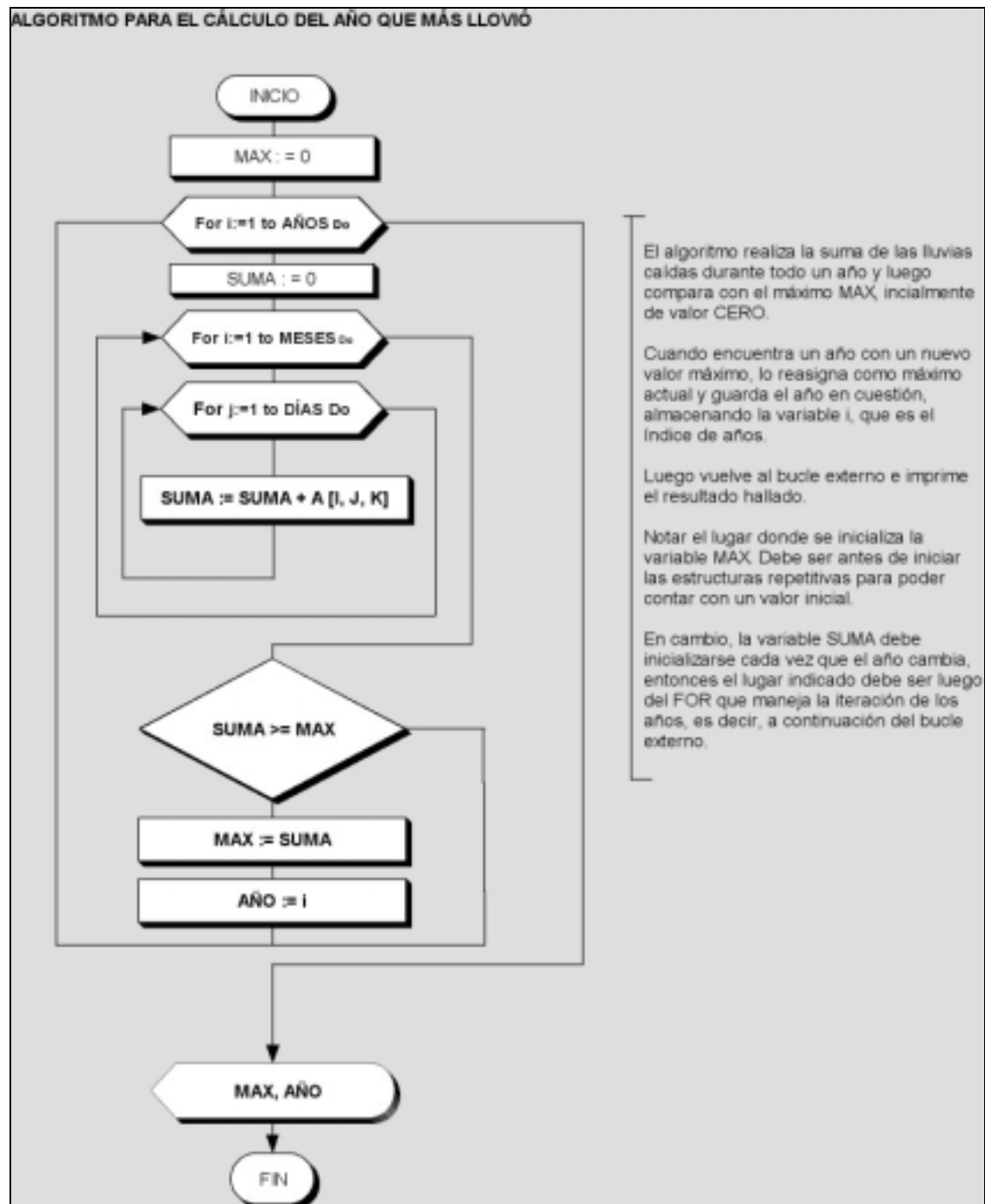
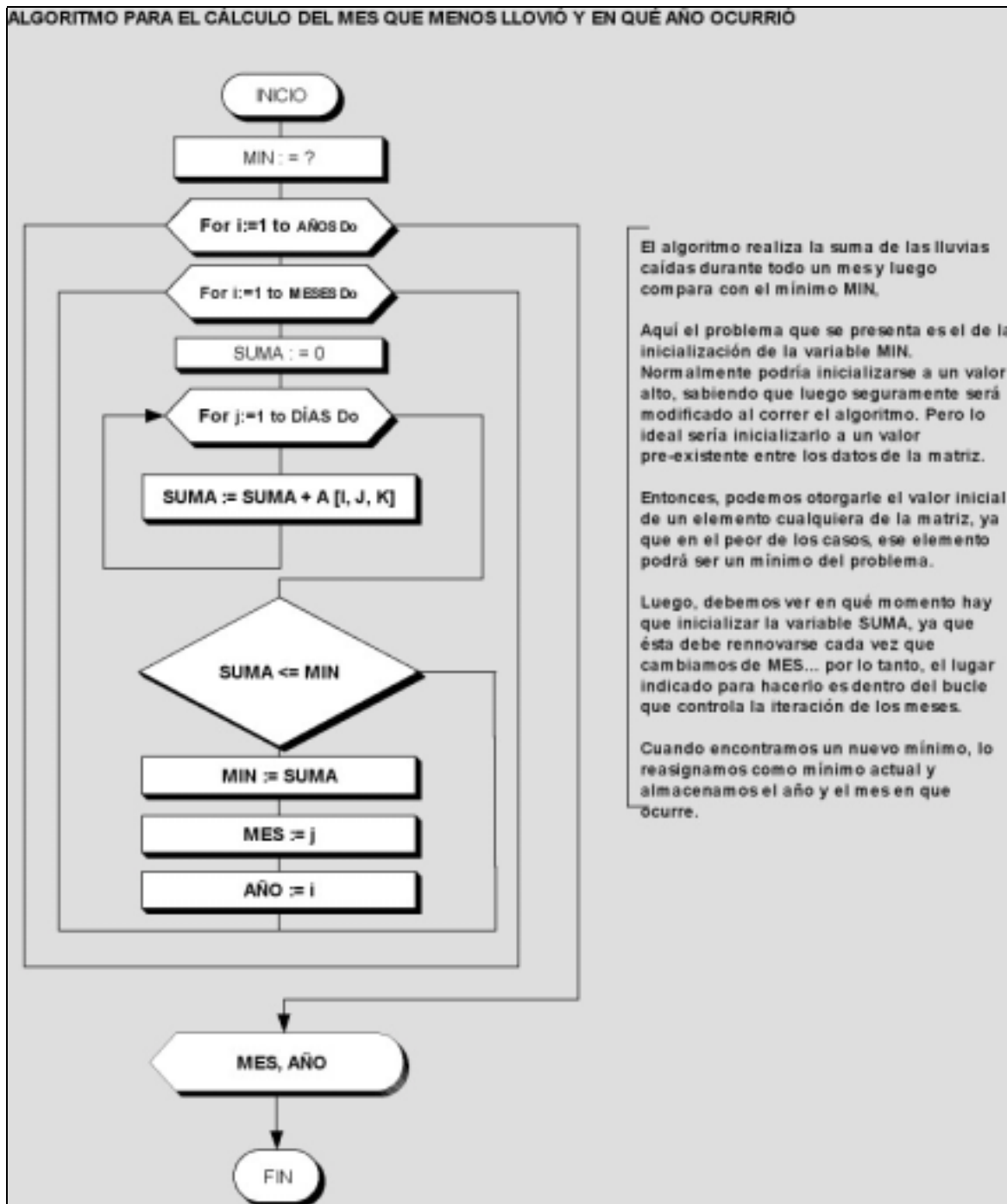
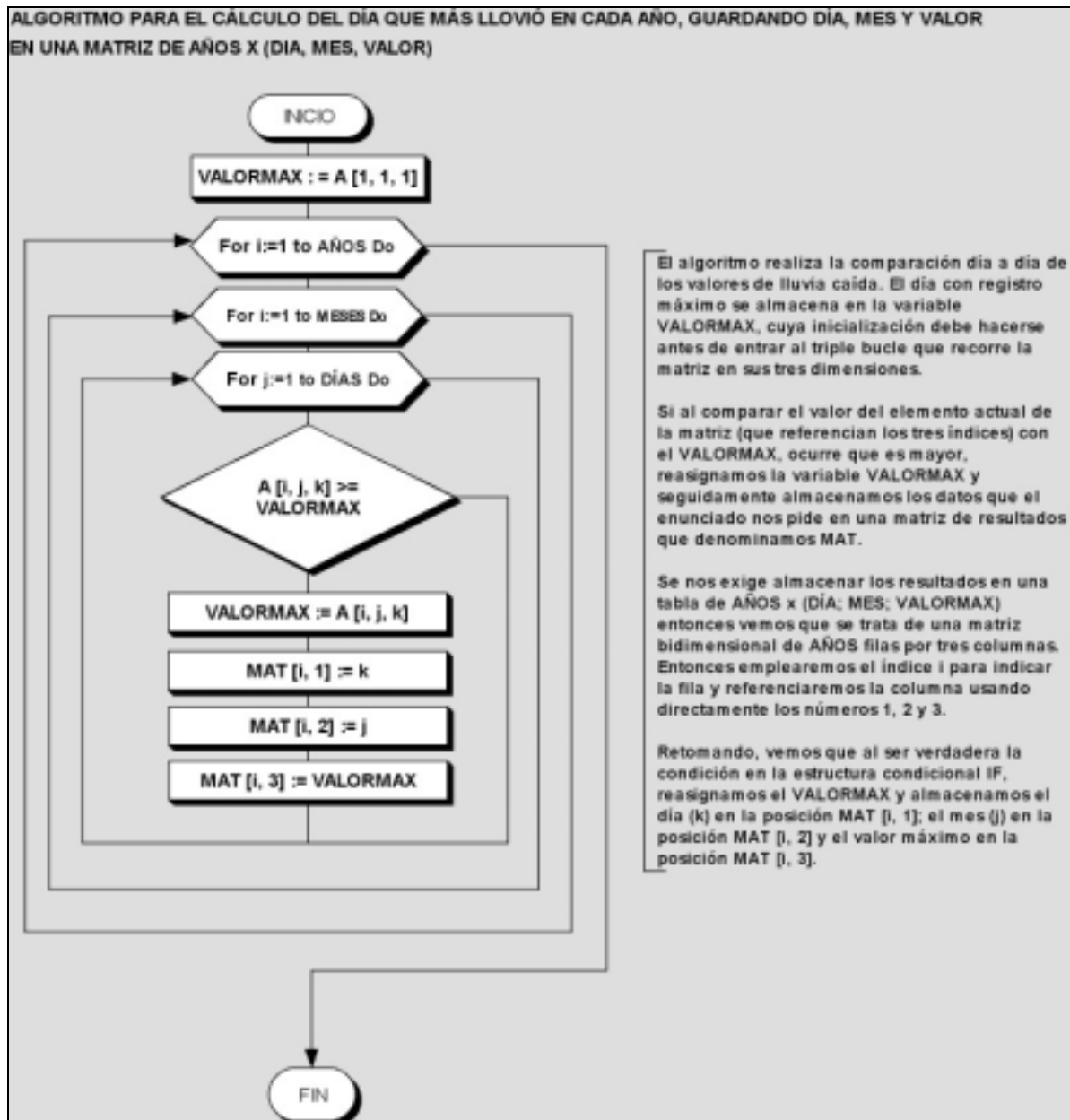
DIAGRAMAS DE FLUJO (A)

DIAGRAMA DE FLUJO (B)

DIAGRAMAS DE FLUJO (C)

DIAGRAMAS DE FLUJO (D)

CÓDIGO IMPLEMENTADO PARA EL ÍTEM (D)

```

program Matriz3D_d;

const
  ANIOS = 10;
  MESES = 12;
  DIAS = 31;

type
  Mat3D = Array [1..ANIOS,1..MESES,1..DIAS] OF INTEGER;
  Mat2D = Array [1..ANIOS,1..3] OF INTEGER;

var
  A: Mat3D;
  MAT: Mat2D;
  VALORMAX: Integer;
  i,j,k: Integer;

begin
  VALORMAX := A[1,1,1];

  for i:=1 to ANIOS do
    for j:=1 to MESES do
      for k:=1 to DIAS do
        begin
          if (A[i,j,k] >= VALORMAX) then
            begin
              VALORMAX:= A[i,j,k];
              MAT[i,1] := k;
              MAT[i,2] := j;
              MAT[i,3] := VALORMAX;
            end;
          end;
        Readln;
      end.

```

NOTAS:

Como podemos observar hemos definido algunas constantes para facilitar la redacción del código y para facilitar futuras tareas de modificación. También hemos definido dos nuevos tipos de datos: Mat3D y Mat2D, para poder trabajar leyendo los datos de la primera y guardando los datos requeridos en la segunda.

Dentro del cuerpo del programa, vemos la inicialización de la variable entera VALORMAX al valor del primer elemento de la matriz tridimensional **A**. Luego abrimos los tres bucles FOR, uno para cada dimensión de la matriz y dentro del bucle más interno realizamos la comparación del valor del elemento que estamos recorriendo respecto de VALORMAX.

Según sea verdadera esta comparación, reasignamos VALORMAX y guardamos en la matriz bidimensional **MAT** los datos que nos pide el enunciado, en el formato AÑOS x (DÍA, MES, VALOR). Veamos la estructura de esta matriz de salida:

AÑOS	DÍA	MES	VALOR
i	MAT[i,1]	MAT[i,2]	MAT[i,3]
1	k	j	VALORMAX
.	.	.	.
.	.	.	.
.	.	.	.
10	k	j	VALORMAX

Es decir, para el año (i) se registró VALORMAX de lluvia caída, y ocurrió el día MAT[i,1] en el mes MAT[i,2].